



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/710,921	11/09/2000	Rick Allen Hamilton II	AUS9-2000-0561-US1	5545
35525	7590	06/05/2006	EXAMINER	
IBM CORP (YA) C/O YEE & ASSOCIATES PC P.O. BOX 802333 DALLAS, TX 75380			DUONG, THOMAS	
			ART UNIT	PAPER NUMBER
			2145	

DATE MAILED: 06/05/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/710,921	Applicant(s) HAMILTON ET AL.	
	Examiner Thomas Duong	Art Unit 2145	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 14 March 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 2-10, 12-20, and 22-33 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 2-10, 12-20, and 22-33 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date <u>5/3/01</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Response to Amendment

1. This office action is in response to the applicants Amendment filed on March 14, 2006.
Claims 2-10, 12-20, and 22-33 are presented for further consideration and examination.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.
3. Claims 2, 12, and 22 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.
4. With regards to claims 2, 12, and 22, Applicants recite the limitation,
 - "said first one of said plurality of commands"There is insufficient antecedent basis for this limitation in the claims.
5. With regards to claims 2, 12, and 22, Applicants recite the limitation,
 - "said second one of said plurality of commands"There is insufficient antecedent basis for this limitation in the claims.

Claim Rejections - 35 USC § 103

Art Unit: 2145

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 2-10, 12-20, and 22-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Neches et al. (US004925311) and in view of Blumenau (US006018779A).

8. With regard to claims 2, 12, and 22, Neches discloses,

- *selecting said plurality of commands from the environment which executes commands concurrently;* (Neches, col.3, line 36 – col.4, line 40)

Neches discloses, *"systems and methods in accordance with the invention [to] carry out each of a number of tasks by establishing arbitrary subset groupings or partitions within a number of parallel processors"* (Neches, col.3, lines 36-39). In addition, Neches discloses, *"each processor carrying out the assigned steps for its tasks asynchronously with respect to others"* (Neches, col.3, lines 43-45).

Neches also discloses, *"an originating processor identifies a specific task by parsing individual steps... Performance of the task effectively begins when steps are distributed, as by hashing algorithm, to the responsible processors, which verify the destinations and concurrently identify themselves as members of the group. They then can process the step assigned to them, using the dynamic group location to update local task status, test for the status of other processors involved, eliminate conflicts and ultimately either abort the transaction or normally*

relinquish the dynamic group in the process" (Neches, col.4, lines 27-40).

Hence, Neches teaches of a distributed environment, wherein subset groupings or partitions within a number of parallel processors are established, and that each processor carries out the assigned steps for its task asynchronously with respect to others.

- *scheduling execution of said selected plurality of commands in a programming order, said scheduling step comprising:* (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21; col.9, lines 22-45)

Neches discloses, *"the individual processors can validate their subtasks and insure orderly performance by using the semaphores to block interfering commands and to test state of other processors in the group. An individual processor updates the local status word as its subtasks are carried out and it progresses toward a final state of relinquishment, or indicates an abort condition"* (Neches, col.3, line 67 – col.4, line 6). Hence, Neches teaches of ordering the steps of the individual tasks to insure orderly performance.

- *beginning processing of said first process;* (Neches, col.8, line 36 – col.9, line 21)

Neches teaches of *"the AMP should to Begin Transaction (BT) processing. The message may (and typically does) indicate other processing that should be performed by the AMP after successfully completing Begin Transaction processing"* (Neches, col.8, lines 42-46) and *"commences processing the subtask"* (Neches, col.8, line 59). Hence, Neches teaches of processing the subtask, which includes a subset of tasks or commands.

- *executing said first one of said plurality of commands in response to said beginning processing of said first process, wherein said first one of said plurality of commands executes only while said first process is executing;*
(Neches, col.8, line 36 – col.9, line 21)

Neches teaches of *"the AMP should to Begin Transaction (BT) processing. The message may (and typically does) indicate other processing that should be performed by the AMP after successfully completing Begin Transaction processing"* (Neches, col.8, lines 42-46) and *"commences processing the subtask"* (Neches, col.8, line 59). In addition, Neches discloses, *"each processor carrying out the assigned steps for its tasks asynchronously with respect to others"* (Neches, col.3, lines 43-45). Neches also discloses, *"an originating processor identifies a specific task by parsing individual steps... Performance of the task effectively begins when steps are distributed, as by hashing algorithm, to the responsible processors, which verify the destinations and concurrently identify themselves as members of the group. They then can process the step assigned to them, using the dynamic group location to update local task status, test for the status of other processors involved, eliminate conflicts and ultimately either abort the transaction or normally relinquish the dynamic group in the process"* (Neches, col.4, lines 27-40). Hence, Neches teaches of a distributed environment, wherein subset groupings or partitions within a number of parallel processors are established, and that each processor carries out the assigned steps for its task asynchronously with respect to others until it is finished or is forced to abort.

Art Unit: 2145

- *and beginning processing of said second process only in response to a completion of processing of said first process. (Neches, col.8, line 36 – col.9, line 21)*

Neches teaches of the first AMP can begin processing the next subset of tasks or subtask following completion of its first subtask. The AMPs will continue this process until there are no more subtasks to be processed and they will make themselves available for a subsequent task or transaction.

However, Neches does not explicitly disclose,

- *encapsulating said first one of said plurality of commands in a first process and encapsulating said second one of said plurality of commands in a second process;*

Blumenau teaches,

- *encapsulating said first one of said plurality of commands in a first process and encapsulating said second one of said plurality of commands in a second process; (Blumenau, col.1, line 41 – col.2, line 34; col.7, lines 1-14)*

Blumenau discloses, *“a method of sending a plurality of commands to a remote device which executes the plurality of commands. The method includes the steps of encapsulating the plurality of commands within a single command; and sending the single command to the remote device”* (Blumenau, col.1, lines 41-46). In addition, Blumenau discloses, *“it grabs the commands that are being held within the I/O queue on the host side (step 102), encapsulates that block of commands within the appropriate encapsulated SCSI command (step 106), and sends that encapsulated SCSI command to the data storage system during the data phase (step 106)”* (Blumenau, col.7, lines 5-11) to process the embedded

commands. Hence, Blumenau teaches of encapsulating a plurality of commands (i.e., Applicants' said plurality of commands) within a single command (i.e., Applicants' process) and processing that single command, which contains plurality of encapsulated commands.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Blumenau with the teachings of Neches to reduce the *"undesired level of overhead activity and processor time committed to overhead functions"* (Neches, col.2, lines 44-45). According to Neches, *"consequently, there is a need for capability for partitioning multiple concurrent subgroupings of processors in a system which can also eliminate conflicting states and which can dynamically change these groupings as needed"* (Neches, col.2, lines 60-64). And, according to Blumenau, *"an advantage of the invention is that the power of the communication protocol is substantially increased..."* (Blumenau, col.2, lines 27-28) by encapsulating a plurality of commands into a single SCSI command, thus reducing the number of arbitrations that are necessary for sending the set of I/O commands to be processed.

9. With regard to claims 3, 13, and 23, Neches and Blumenau disclose,
 - *further comprising the step of completing processing of said first process in response to a completion of execution of said first one of said plurality of commands.* (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21; Blumenau, col.1, line 41 – col.2, line 34)
10. With regard to claims 4, 14, and 24, Neches and Blumenau disclose,

- *further comprising the step of executing said second one of said plurality of commands in response to said beginning processing of said second process* (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21; Blumenau, col.1, line 41 – col.2, line 34)
11. With regard to claims 5, 7, 15, 17, 25, and 27, Neches and Blumenau disclose,
- *further comprising the step of determining whether said first process is currently executing.* (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21)
Neches teaches of “at the completion of commit processing, the AMP tests for whether it is the last done, querying the other members of the dynamic group” (Neches, col.9, lines 11-13).
 - *wherein said step of determining whether said first process is currently executing further comprises the steps of:*
 - *assigning a first process identifier to said first process; and* (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21)
Neches discloses, “the individual processors can validate their subtasks and insure orderly performance by using the semaphores to block interfering commands and to test state of other processors in the group. An individual processor updates the local status word as its subtasks are carried out and it progresses toward a final state of relinquishment, or indicates an abort condition” (Neches, col.3, line 67 – col.4, line 6). Hence, Neches teaches of sending updates as its steps/subtasks are carried out while progressing toward a final state of relinquishment.

- *utilizing said first process identifier to determine whether said first process is currently executing.* (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21)

Neches discloses, *“the individual processors can validate their subtasks and insure orderly performance by using the semaphores to block interfering commands and to test state of other processors in the group. An individual processor updates the local status word as its subtasks are carried out and it progresses toward a final state of relinquishment, or indicates an abort condition”* (Neches, col.3, line 67 – col.4, line 6). Hence, Neches teaches of sending updates as its steps/subtasks are carried out while progressing toward a final state of relinquishment.

12. With regard to claims 6, 16, and 26, Neches and Blumenau disclose,

- *wherein said step of determining whether said first process is currently executing further comprises the steps of:*
 - *establishing a return code variable; and* (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21; col.9, lines 46-58; Blumenau, col.1, line 41 – col.2, line 34)

Neches discloses, *“the individual processors can validate their subtasks and insure orderly performance by using the semaphores to block interfering commands and to test state of other processors in the group. An individual processor updates the local status word as its subtasks are carried out and it progresses toward a final state of relinquishment, or indicates an abort condition”* (Neches, col.3, line 67 – col.4, line 6). Hence, Neches teaches of

sending updates as its steps/subtasks are carried out while progressing toward a final state of relinquishment.

- *utilizing said return code variable to indicate whether said first process is currently executing.* (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21; col.9, lines 46-58; Blumenau, col.1, line 41 – col.2, line 34)
Neches discloses, *“the individual processors can validate their subtasks and insure orderly performance by using the semaphores to block interfering commands and to test state of other processors in the group. An individual processor updates the local status word as its subtasks are carried out and it progresses toward a final state of relinquishment, or indicates an abort condition”* (Neches, col.3, line 67 – col.4, line 6). Hence, Neches teaches of sending updates as its steps/subtasks are carried out while progressing toward a final state of relinquishment.

13. With regard to claims 8-9, 18-19, and 28-29, Neches and Blumenau disclose,

- *further comprising the step of:*
 - *searching a process table for said first process identifier* (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21; col.9, lines 46-58; Blumenau, col.1, line 41 – col.2, line 34)
 - *determining that said first process is executing in response to locating said process identifier is said process table; and* (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21; col.9, lines 46-58; Blumenau, col.1, line 41 – col.2, line 34)

Art Unit: 2145

- *determining that said first process is not executing in response to a failure to locate said process identifier in said process table (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21; col.9, lines 46-58; Blumenau, col.1, line 41 – col.2, line 34)*
 - *further comprises the steps of:*
 - *setting said return code variable equal to a first value while said first process is executing; and (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21; col.9, lines 46-58; Blumenau, col.1, line 41 – col.2, line 34)*
 - *setting said return code variable equal to a second value when said first process has completed executing. (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21; col.9, lines 46-58; Blumenau, col.1, line 41 – col.2, line 34)*
14. With regard to claims 10, 20, and 30, Neches and Blumenau disclose,
- *further comprising the steps of:*
 - *establishing a timer for said first process; (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21; col.9, lines 46-58; Blumenau, col.1, line 41 – col.2, line 34)*
 - *starting said timer in response to executing said first process; and (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21; col.9, lines 46-58)*
 - *testing said return code variable to determine whether said return code variable is equal to said second value upon the expiration of said timer. (Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21; col.9, lines 46-58; Blumenau, col.1, line 41 – col.2, line 34)*

15. With regard to claims 31-33, Neches and Blumenau disclose,
- *Wherein said first process and said second process are included in a script*
(Neches, col.3, line 36 – col.4, line 40; col.8, line 36 – col.9, line 21; col.9, lines 46-58; Blumenau, col.1, line 41 – col.2, line 34)

Response to Arguments

16. Applicant's arguments with respect to *claims 2-10, 12-20, and 22-33* have been considered but they are not persuasive.
17. With regard to claim 2, the Applicants point out that:
- *Applicants urge that this is classic hindsight analysis, where the Examiner is using Applicants' own patent specification as a blueprint to modify the teachings of the cited references in accordance with the claimed invention... Thus, a person of ordinary skill in the art would not have been motivated to modify the teachings of Blumenau in accordance with the features recited in Claim 2. This further evidence that the only motivation to modify the teachings of the cited references in accordance with present invention must be coming from the present invention itself, which is improper hindsight analysis.*
 - *There is simply no suggestion or other motivation per the teachings of Neches to modify such teachings to include the claimed command encapsulation technique, due to Neches desire to streamline processing in a multi-processing environment (col.2, lines 37-53).*

However, the Examiner finds that the Applicants' arguments are not persuasive because it would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Blumenau with the teachings of Neches to reduce the *"undesired level of overhead activity and processor time committed to overhead functions"* (Neches, col.2, lines 44-45). According to Neches, *"consequently, there is a need for capability for partitioning multiple concurrent subgroupings of processors in a system which can also eliminate conflicting states and which can dynamically change these groupings as needed"* (Neches, col.2, lines 60-64). And, according to Blumenau, *"an advantage of the invention is that the power of the communication protocol is substantially increased..."* (Blumenau, col.2, lines 27-28) by encapsulating a plurality of commands into a single SCSI command, thus reducing the number of arbitrations that are necessary for sending the set of I/O commands to be processed. Furthermore, the Applicants clearly admit that *"[further], the cited Neches reference is keen on reducing overhead associated with a multiprocessor system"* (Remarks, para.3) and *"Blumenau expressly states that a plurality of commands are encapsulated in a single command to reduce processing times, and in particular to reduce processing overhead associated with arbitration that is required"* (Remarks, para.2).

18. With regard to claim 2, the Applicants point out that:

- *The cited reference does not teach/suggest, nor has the Examiner alleged any such teaching/suggestion of, the claimed step of "executing said first one of said plurality of commands in response to said beginning processing of said first*

process, wherein said first one of said plurality of commands executes only while said first process is executing”.

However, the Examiner finds that the Applicants' arguments are not persuasive because Neches discloses, *“each processor carrying out the assigned steps for its tasks asynchronously with respect to others”* (Neches, col.3, lines 43-45). Neches also discloses, *“an originating processor identifies a specific task by parsing individual steps... Performance of the task effectively begins when steps are distributed, as by hashing algorithm, to the responsible processors, which verify the destinations and concurrently identify themselves as members of the group. They then can process the step assigned to them, using the dynamic group location to update local task status, test for the status of other processors involved, eliminate conflicts and ultimately either abort the transaction or normally relinquish the dynamic group in the process”* (Neches, col.4, lines 27-40). Hence, Neches teaches of a distributed environment, wherein subset groupings or partitions within a number of parallel processors are established, and that each processor carries out the assigned steps for its task asynchronously with respect to others until it is finished or is forced to abort.

19. With regard to claim 2, the Applicants point out that:

- *The cited Blumenau reference, which is being used as teaching this claimed feature, teaches that a plurality of commands are encapsulated into a single command (see, e.g., Blumenau's Abstract). Claim 2 is specifically directed to encapsulating a single command in a process. A teaching of encapsulating multiple commands into a single command does not teach or otherwise suggest*

encapsulating anything into a process, as expressly recited in Claim 2. Because Claim 2 recites both a command a process, they are not the same thing and therefore it is error to equate Blumenau's command with the claimed process.

However, the Examiner finds that the Applicants' arguments are not persuasive because Blumenau discloses, *"a method of sending a plurality of commands to a remote device which executes the plurality of commands. The method includes the steps of encapsulating the plurality of commands within a single command; and sending the single command to the remote device"* (Blumenau, col.1, lines 41-46). In addition, Blumenau discloses, *"it grabs the commands that are being held within the I/O queue on the host side (step 102), encapsulates that block of commands within the appropriate encapsulated SCSI command (step 106), and sends that encapsulated SCSI command to the data storage system during the data phase (step 106)"* (Blumenau, col.7, lines 5-11) to process the embedded commands. Furthermore, Blumenau discloses, *"encapsulating I/O commands into a single SCSI command reduces the number of arbitrations that are necessary for sending the set of I/O commands to the peripheral device"* (Blumenau, col.2, lines 21-24). Hence, Blumenau teaches of encapsulating a plurality of I/O commands (i.e., Applicants' said plurality of commands) within a single SCSI command (i.e., Applicants' process) and processing that single SCSI command, which contains plurality of encapsulated I/O commands so *"that the power of the communication protocol is substantially increased"* (Blumenau, col.2, lines 27-28).

20. With regard to claim 2, the Applicants point out that:

Art Unit: 2145

- *The cited Blumenau reference, which is being used as teaching this claimed feature, teaches that a plurality of commands are encapsulated into a single command (see, e.g., Blumenau's Abstract). Claim 2 is specifically directed to encapsulating a single command in a process. A teaching of encapsulating multiple commands into a single command does not teach or otherwise suggest encapsulating anything into a process, as expressly recited in Claim 2. Because Claim 2 recites both a command a process, they are not the same thing and therefore it is error to equate Blumenau's command with the claimed process.*

However, the Examiner finds that the Applicants' arguments are not persuasive because the scope of the language *claim 2* does not exclude the multiplicity of commands from being encapsulated into a single command as argued by the Applicants.

21. With regard to claim 6, the Applicants point out that:

- *Further with respect to Claim 6 (and dependent Claim 7), it is urged that none of the cited references teach or suggest the claimed feature of "wherein said step of determining whether said first process is currently executing further comprises the steps of establishing a return code variable for the first process; and utilizing said return code variable to indicate whether said first process is currently executing".*

However, the Examiner finds that the Applicants' arguments are not persuasive because Neches discloses, "*the individual processors can validate their subtasks and insure orderly performance by using the semaphores to block interfering commands and to test state of other processors in the group. An individual*

processor updates the local status word as its subtasks are carried out and it progresses toward a final state of relinquishment, or indicates an abort condition" (Neches, col.3, line 67 – col.4, line 6). Hence, Neches teaches of sending updates as its steps/subtasks are carried out while progressing toward a final state of relinquishment.

22. With regard to claim 7, the Applicants point out that:

- *Further with respect to Claim 7 (and dependent Claims 8-10), it is urged that none of the cited references teach or suggest the claimed feature of "utilizing said first process identifier to determine whether said first process is currently executing".*

However, the Examiner finds that the Applicants' arguments are not persuasive because Neches discloses, *"the individual processors can validate their subtasks and insure orderly performance by using the semaphores to block interfering commands and to test state of other processors in the group. An individual processor updates the local status word as its subtasks are carried out and it progresses toward a final state of relinquishment, or indicates an abort condition"* (Neches, col.3, line 67 – col.4, line 6). Hence, Neches teaches of sending updates as its steps/subtasks are carried out while progressing toward a final state of relinquishment.


Art Unit: 2145

Conclusion

23. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a). A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.
24. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Thomas Duong whose telephone number is 571/272-3911. The examiner can normally be reached on M-F 7:30AM - 4:00PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jason D. Cardone can be reached on 571/272-3933. The fax phone numbers for the organization where this application or proceeding is assigned are 571/273-8300 for regular communications and 571/273-8300 for After Final communications.

Thomas Duong (AU2145)

May 25, 2006



Jason D. Cardone

Supervisory PE (AU2145)